

5 Application Infrastructure

This section discusses different types of application used in a control system. It also gives requirements on the operational environment, data protocols, and internal functionality common to all applications.

5.1 Types of Applications

There are three types of applications in the control system:

1. **Low-Level Applications** (described in [3]), which acquire data from physical equipment, front-ends, or other external sources. This section includes one kind of low-level applications, the **Open Access Clients (OAC)**, also known as *software front-ends*.
2. **High-Level Applications** (described in [6]), which provide human-readable data to the end users. The two types of high-level applications are:
 - **Custom Applications**, programs designed specifically for use in the current system.
 - **Third-Party Applications**, general-purpose programs that integrate with the rest of the system through well-defined protocols or data formats. This includes web browsers, MATLAB, JAS, and other tools routinely used to visualize and process data.
3. **Middleware**, which mediates the communication between the low-level and the high-level applications and provides additional services to both. The middleware includes:
 - **Central Services** (discussed in section 4).
 - **Daemons**, periodically or continuously running routines.
 - **Server-Side Parts** of multi-tier high-level applications.
 - **Web Applications**, providing data via HTTP.*

The applications may run on three types of computer nodes:

1. **Client Nodes**, users' PCs, laptops, etc.; as well as shared consoles in control rooms. The client nodes can be used to execute only high-level applications.
2. **Central Nodes**, servers that are administered and controlled by the Controls Department and accessible by the users via remote tools. The central nodes can be used to execute high-level applications, the middleware, and Open Access Clients. As all server-side programming components are headless, they shall run under the control of a specialized application environment (an application server), rather than a human being.
3. **Front-End Nodes**, servers controlled by the Control Department and generally not accessible by the users. The front-end nodes can be used to execute only low-level applications. Specific requirements are given in section 3.

* A Web Application is a server-side component for a web browser, but as this technology becomes ubiquitous, Web Applications are viewed as an individual class of applications. Without a graphical user interface.

No.	Requirement	Source	Priority
CXR-AI-10	The control system shall provide a technical specification for the hard- and software environment sufficient to run high-level applications.	A. Petrov 01-2008	Critical
CXR-AI-20	The control system shall enable high-level applications on the central nodes.	A. Petrov 01-2008	Expected
CXR-AI-25	For high-level applications running on the central nodes, the control system shall provide a means to display their GUI on the client nodes.	A. Petrov 01-2008	Expected
CXR-AI-30	The control system shall include a set of central nodes capable of running high-level applications.	A. Petrov 01-2008	Expected
CXR-AI-35	The control system shall enable a subset of high-level applications on the client nodes.	A. Petrov 01-2008	Expected
CXR-AI-40	The control system shall provide a tool for the users to select and launch high-level applications (see also CXR-HL-201).	A. Petrov 01-2008	Critical
CXR-AI-50	The use of third-party high-level applications shall be approved by a board of experts. Unapproved programs shall <u>not</u> be supported.	C. Schumann 01-2008	Critical
CXR-AI-60	The control system shall provide an application server for the server-side programming components.	A. Petrov 01-2008	Critical
CXR-AI-70	The application server shall be capable of executing Central Services.	A. Petrov 01-2008	Critical
CXR-AI-80	The application server shall be capable of executing Open Access Clients.	A. Petrov 01-2008	Critical
CXR-AI-90	The application server shall be capable of executing deamons, which are continuously or periodically running headless routines.	A. Petrov 01-2008	Expected
CXR-AI-100	The application server shall be capable of executing server-side parts of multi-tier high level applications.	A. Petrov 01-2008	Expected
CXR-AI-110	The application server shall be capable of executing web applications.	A. Petrov 01-2008	Expected
CXR-AI-120	It shall be possible to start, stop, and reconfigure individual components inside the application server without affecting others.	A. Petrov 01-2008	Expected
CXR-AI-130	The application server shall be able to recover the states of all the components upon restart or after spontaneous failures.	A. Petrov 01-2008	Critical

CXR-AI-140	The control system shall support individual configuration of application servers for each central node.	A. Petrov 01-2008	Critical
CXR-AI-150	The application server shall report its internal state through the Data Acquisition Service [4.2].	A. Petrov 01-2008	Expected
CXR-AI-160	The application server shall provide a mechanism to collect and keep application log files.	A. Petrov 01-2008	Expected

5.2 Application Protocols

The application protocols are used by high-level applications to communicate with the middleware and low-level components. They are also used by the server-side components to communicate with each other. The right choice of application protocols can facilitate the development of high-level applications and improve integration with third-party software.

The requirements in this section use a terminology from the *OSI Reference Model*, described at http://en.wikipedia.org/wiki/osi_model.

No.	Requirement	Source	Priority
CXR-AI-170	The application protocols shall be platform and language independent.	A. Petrov 01-2008	Critical
CXR-AI-180	The control system shall specify the protocols that will be supported.	A. Petrov 01-2008	Expected
CXR-AI-190	Each application protocol shall be implemented with a binary presentation layer.	A. Petrov 01-2008	Desired
CXR-AI-200	Each application protocol shall be implemented with a human-readable presentation layer.	A. Petrov 01-2008	Desired
CXR-AI-210	All presentation-layer data formats shall be explicitly specified.	A. Petrov 01-2008	Expected
CXR-AI-230	The inbound connections shall be accepted on well-defined ports.	A. Petrov 01-2008	Expected

5.3 General-Purpose Database

No.	Requirement	Source	Priority
CXR-AI-900	The control system shall provide a general-purpose application database, accessible from, at least, the central nodes.	A. Petrov 01-2008	Critical
CXR-AI-910	The control system shall implement database connection pools. All applications shall obtain database connections from the pools.	A. Petrov 01-2008	Expected

CXR-AI-920	The database shall support, but not mandate, client authentication.	A. Petrov 01-2008	Critical
CXR-AI-930	The database shall mandate authorization of clients modifying the data.	A. Petrov 01-2008	Critical
CXR-AI-940	Common dictionaries of objects shall use the Naming Service [4.1], rather than the application database.	A. Petrov 01-2008	Expected
CXR-AI-950	High-level applications shall <u>not</u> utilize direct database access from the client nodes. To access the database, the client-side applications shall use a two-tier approach, when the DB connection is opened in the middle tier.	A. Petrov 01-2008	Expected
CXR-AI-960	All central services and high-level applications shall be designed in a way that makes them reasonably independent from any particular database implementation (e.g., by using Service Provider Interfaces).	A. Petrov 01-2008	Desired
CXR-AI-970	The control system shall keep a log of application database queries.	K. Cahill 01-2008	Expected

5.4 Security

No.	Requirement	Source	Priority
CXR-AI-240	The control system shall have a written security policy.	A. Petrov 01-2008	Critical
CXR-AI-250	The control system shall implement a central Identity Database, which registers credentials of the principals (users).	A. Petrov 01-2008	Critical
CXR-AI-260	At a minimum, each credential shall include a unique name of the principal, a principal type, an optional description, and a set of assigned roles (named permissions).	A. Petrov 01-2008	Expected
CXR-AI-270	The data in the Identity Database (list of user and roles) shall be available through the Naming Service [4.1].	A. Petrov 01-2008	Expected
CXR-AI-280	The Identity database shall include a graphical user interface (GUI) for remote configuration and monitoring.	A. Petrov 01-2008	Expected
CXR-AI-290	The control system shall support, but not mandate, strong authentication of user principals that satisfies the requirements of the lab security policy.	A. Petrov 01-2008	Critical
CXR-AI-300	The control system shall support a mechanism of Single Sign-On for user principals.	A. Petrov 01-2008	Expected

CXR-AI-310	The control system shall support principals assigned to computer nodes and programming processes, and provide appropriate mechanisms of their authentication.	A. Petrov 01-2008	Expected
CXR-AI-320	The control system shall provide a mechanism of default authentication on shared consoles.	A. Petrov 01-2008	Critical
CXR-AI-325	A client program shall be able to identify itself during authentication (e.g., by providing a high-level application name, OAC name, etc). This information shall be used only in addition to a trusted form of authentication.	T. Zingelman 01-2008	Expected
CXR-AI-330	The control system shall keep a log of all authentication attempts.	A. Petrov 01-2008	Expected
CXR-AI-340	The application protocols [5.2] shall incorporate, in the transport layer, a cryptographic protocol providing client authentication and protection against message tampering.	A. Petrov 01-2008	Desired
CXR-AI-350	Authorization decisions shall be made by the server-side components or front-ends based on roles assigned to the current principal.	A. Petrov 01-2008	Expected
CXR-AI-360	A middleware component shall be able to delegate client credentials to another middleware or low-level component.	A. Petrov 01-2008	Desired

5.5 Application Framework

No.	Requirement	Source	Priority
CXR-AI-370	The control system shall provide a programming framework for standard high-level applications.	A. Petrov 01-2008	Expected
CXR-AI-380	The application framework shall provide a graphical user interface with a common look-and-feel [CXR-HL-100].	A. Petrov 01-2008	Expected
CXR-AI-390	The application framework shall provide an API to the Central Services.	A. Petrov 01-2008	Expected
CXR-AI-400	The application framework shall implement the GUI features required by the Central Services [CXR-CS-680, CXR-CS-690, CXR-CS-940].	A. Petrov 01-2008	Expected
CXR-AI-410	The application framework shall incorporate an API to the application database [5.3], and a database connection pool.	A. Petrov 01-2008	Expected
CXR-AI-420	The application framework shall test the availability of required external resources (Central Services, application database, and server-side components). If a resource is	A. Petrov 01-2008	Expected

	not reachable, the framework shall notify the user and disable the entire application.		
CXR-AI-430	The application framework shall send email notifications to application subscribers should the program terminate abnormally.	C. Schumann 01-2008	Expected
CXR-AI-440	Each application shall have a link to an online help page.	A. Petrov 01-2008	Expected